Accepted Publication to the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)
November 4-7, 2018, Maui, Hawaii, USA
DOI: 10.1109/ITSC.2018.8569323

LLama-SLAM: Learning High-Quality Visual Landmarks for Long-Term Mapping and Localization

Stefan Luthardt*, Volker Willert*, Jürgen Adamy*

***Abstract*— The precise localization of vehicles is an important requirement for autonomous driving or advanced driver assistance systems. Using common GNSS the ego position can be measured but not with the reliability and precision necessary. An alternative approach to achieve precise localization is the usage of visual landmarks observed by a camera mounted in the vehicle. However, this raises the necessity of reliable visual landmarks that are easily recognizable and persistent. We propose a novel SLAM algorithm that focuses on learning and mapping such visual long-term landmarks (LLamas). The algorithm therefore processes stereo image streams from several recording sessions in the same spatial area. The key part within LLama-SLAM is the assessment of the landmarks with quality values that are inferred as viewpoint dependent probabilities from observation statistics. By adding solely landmarks of high quality to the final LLama Map, it can be kept compact while still allowing reliable localization. Due to the long-term evaluation of the GNSS measurement during the sessions, the landmarks can be positioned precisely in a global referenced coordinate system. For a first assessment of the algorithm's capabilities, we present some experimental results from the mapping process combining three sessions recorded over two months on the same route.**

I. Introduction

A. Motivation

To enable autonomous driving or advanced driver assistance systems a precise estimation of the vehicle's position is essential. The positioning is necessary to use information that is not measured in the current instance of time by the vehicle sensors, like map data, sensor data from past time instances or sensor data received from other vehicles. Such data needs to be positioned precisely relative to the vehicle, which is equivalent to estimating the vehicle's position. Vice versa localization is also necessary to contribute sensor information to a map or to share useful information with other vehicles. In principle GNSS (global navigation satellite systems) like GPS are a good possibility to solve the positioning task. However for more demanding tasks they are not feasible due to their limited precision and availability. Another possibility for localization is the usage of visual landmarks which are observed by a camera mounted in the vehicle. Therefore V-SLAM (Visual Simultaneous Localization and Mapping) algorithms [1]–[3] are employed, which find prominent environment points and use them as landmarks within a self-build map. This is an appealing approach since cameras are already part of many modern cars and in theory high precision is achievable. However, most of current V-SLAMs focus on a short-term usage of these landmarks in one continuous data record and demand a large number of landmarks. In this paper, we introduce a novel concept for visual landmarks that focuses on long-term usage of landmarks which should be more practical for vehicle localization. We developed a new V-SLAM algorithm that processes multiple driving sessions recorded in the same spatial area during a period of days and months. This algorithm evaluates the persistence and appearance of the landmarks over time to identify long-term landmarks (LLamas).

*Control Methods and Robotics, TU Darmstadt, Germany

Fig. 1. LLama-SLAM aims to find high-quality long-term landmarks on persistent structures like the shop sign in the images above. A high-quality landmark is easy to identify, persistent, visible in most conditions from several viewpoints and aids the self-localization. (See Sec. V-A for more landmark examples.)

B. LLamas: Long-Term Landmarks

To explain our concept of a LLama we will use a vivid analogy from human behavior: As a tourist in Paris, France you would probably regard the Eiffel Tower as a landmark, because it:

1) is observable from many places in the city,
2) is easy to identify,
3) is persistent (it stands there since 1889),
4) is visible almost independently of weather or season, and
5) can be used to localize yourself inside Paris.

All these aspects make the Eiffel Tower a great landmark. A LLama in our algorithm is characterized by the same properties as above but in more technical terms. A LLama of high quality should:

1) be observable from several relevant camera positions,
2) be reliably identifiable by its visual descriptor,
3) be persistent, i. e. part of a persistent object,
4) be detectable in several environmental conditions, and
5) aid the localization of the vehicle.

The criteria above influence how often a LLama can be successfully observed and how often it contributes to the vehicle pose estimation. Therefore the "quality" of a LLama can be assessed using an observation statistic. Implicitly the LLama quality thereby also incorporates the goodness of the LLama's position and its descriptor integrity. This

observation based approach is also similar to how humans choose landmarks: If they observe something distinctive, e. g. a tower on a hill, several times on different occasions and find it useful for navigation purposes it becomes a landmark.

In our approach we represent the quality of a LLama as a probabilistic quantity, which is connected to the observation statistic by a conditional probability. Since quality depends on the viewpoint, we record different quality probabilities for different viewpoints in a LLama Quality Map. Therein we also assume a correlation between neighboring viewpoints. The inferred quality values are later used in a selection process and only LLamas with a high quality are added to the map. Due to this quality driven map learning, the map can be kept compact while still offering reliable localization.

C. Paper Overview

In this paper we present our long-term landmark focused V-SLAM algorithm, named LLama-SLAM. We start in the following Sec. II by comparing our approach with existing methods focused on long-term mapping. Afterwards we will explain the basic building blocks of LLama-SLAM in Sec. III. These include graph-based pose estimation and probabilistic quality inference. In Sec. IV we describe the actual mapping procedure within LLama-SLAM. To show the algorithms capabilities we present first experimental results from a long-term experiment in Sec. V. We close by summarizing our work and providing an outlook on future work in Sec. VI.

II. Related Work

A. Robust Long-Term SLAM: An Urgent Problem

Today SLAM is a broad field of research in the robotic and automotive community. Cadena et al. [2] give an up-to-date overview of SLAM methods in general and open problems. They identified long-term robustness and scalability as one of the current major challenges in SLAM research. Furthermore, Cadena et al. highlight the necessity of failure awareness and mechanisms for learning and forgetting. A similar tendency can be found in the SLAM survey for autonomous driving by Bresson et al. [3]. There, accuracy, scalability, availability, recovery, updatability and dynamicity are established as important criteria for SLAMs to enable autonomous driving. According to [3] none of the existing approaches can fulfill these criteria satisfactorily.

Our quality driven landmark mapping approach can be a way to tackle these open problems of long-term SLAM. It allows to deal with false detections or associations, handles long-term changes and provides a criterion for learning and forgetting landmarks. LLama-SLAM is focused on achieving updatability and dynamicity while also leading to improved accuracy and scalability, since only a sparse set of high-quality landmarks represents the final map. Unfortunately, to achieve these improvements, availability has to be sacrificed, i. e. multiple passages of a route are needed, before achieving its full performance. However, we reckon this will not be a big problem in the future, when crowd-sourced data records from nearly all inhabited area will be available.

B. Existing Long-Term Approaches

There are basically two main approaches to tackle long-term changes, both utilizing data from multiple mapping sessions to identify the long-term dynamics. Methods following the first approach aim to capture the major long-term variations either by storing representative measurements [4], [5] or building a model for the variations [6]–[8].

In contrast, the aim in the second approach is filtering out the environment entities that are persistent and reduce the map to those entities. LLama-SLAM belongs to this second category which we consider more favorable in the automotive context. In driving environments there are usually a lot of man-made persistent structures like roads, signs or buildings. These structures allow the creation of a map of persistent landmarks suitable for precise localization. The first approach, i. e. capturing or modeling the variations, demands more storage and is more favorable for highly dynamic environments with very few persistent landmarks, like a park or an office environment. Since methods following the second approach are closer to LLama-SLAM and more popular in the current research, this section will be focused on representative approaches from this category.

Methods focused on long-term mapping basically all rely on observation statistics since this is a quite natural choice as motivated in Sec. I-B. Often the number of observations criterion is combined with measures to ensure good spatial coverage. This is done for example in [9] for surround-view-mapping and in [10] for radar maps. Müllfellner et al. [11] generalize this idea in a framework called Summary Maps. These Summary Maps are obtained by combining data from multiple sessions using a scoring function to assess landmarks usefulness and a sampling policy to select landmarks considering their score and spatial relations. The ideas in the Summary Map framework are related to our notion of a long-term landmark. Dymczyk et al. [12] propose a scoring function for the Summary Map Framework which is based on the rate of observation in relation to the expected observations. This is similar to our idea behind the likelihood formulation in (6). However, we use a more profound probabilistic formulation to link the observations statistics to the landmark quality and include spatial consistency constraints using a Markov Random Field model.

Regarding scoring functions for landmarks there are already some approaches which go beyond the number of observations. Dayoub et al. [13] use a short-term and long-term memory model for the landmarks inspired by the human memory. Based on the observations, landmarks can advance from short-term- to long-term memory or can be forgotten if they have not been observed for a while. Another idea is using probabilistic quantities as landmark score: Johns & Yang [8] learn scene dependent landmark occurrence and co-occurrence probabilities from a set of training sessions. Delobel et al. [14] infer landmark existence in a Bayesian Net using observations statistics and localization validity. Stübler et al. [15] even model the landmarks as Multi-Bernoulli Random Finite Sets which also contain an existence probability.

Comparing all the aforementioned methods to LLama-SLAM three main differences can be identified. Firstly, LLama-SLAM uses inlier feature tracks from an upstream visual odometry system as candidates for the landmarks, i. e. short-term static points are used which already have proven to be useful for self-localization. Secondly, the focus is shifted from camera poses to landmarks by defining the landmark quality as a viewpoint dependent probabilistic property of the landmark which is captured in a local map for each landmark. This novel formulation induces several advantages. The LLama Quality Maps hold quality and viewpoint information combined in a straightforward manner and within these maps the quality correlation between neighboring viewpoints is easily modeled. Since the quality map contains viewpoint information, the camera poses do not have to be stored in the final map as in almost all other methods. Furthermore, the quality map emphasizes and encourages the usage of a landmark from several viewpoints, which allows further reduction of the landmark count. Thirdly, the LLamas' global positions are estimated by combining the GNSS measurements of camera positions from multiple session within the map optimization. This increases the usability of the map and the localization information, since maps and other information can be used and shared in a common coordinate frame.

III. Basic Elements of LLama-SLAM

A. Pose Graph Optimization

Like all SLAM-problems the main goal of our algorithm is to determine the camera poses and the positions of the objects in the environment as precise as possible, based on the given observations. This can be formulated and solved as an optimization problem, where the unknown parameter set $\mathbf{X}$, containing all unknown poses $\mathbf{x}_i$, is optimized until it fits the observations best [16]. This can be expressed as

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{f_{ij} \in \mathrm{C}} f_{ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right) \tag{1}$$

where the sum of all constraint functions f_{ij} from the set of constraints C should be minimized. Each function f_{ij} expresses a constraint between two parameter subsets $\mathbf{x}_i$ and $\mathbf{x}_j$ induced by an observation $\mathbf{z}_{ij}$ and is given by

$$f_{ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right) = \mathbf{e}_{ij}^{\mathrm{T}}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right) \boldsymbol{\Omega}_{ij}\, \mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right). \tag{2}$$

where $\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right)$ is the difference between estimation and measurement, and $\boldsymbol{\Omega}_{ij}$ is the Fisher-Information-Matrix of the observation $\mathbf{z}_{ij}$ specifying its uncertainty. This special structure of the problem is usually expressed in form of a graph with the parameter subsets $\mathbf{x}_i$, $\mathbf{x}_j$ as nodes and the constraint functions $f_{ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right)$ as edges. Fig. 2 shows a simple example of such a graph.

A solution for the optimization problem given by (1) can be numerically found by using the Levenberg-Marquardt algorithm. This can be done efficiently if the special graph structure of the problem is exploited since many entries of the Jacobian-Matrix are actually zero and do not need to be evaluated. We use the g^{2}o-framework from Kümmerle et al. [17] to solve the pose estimation problems in our algorithm. This framework allows to define the optimization problem conveniently as a graph and compute a solution efficiently.

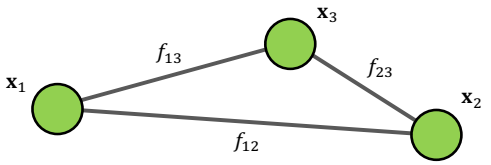

Fig. 2. Simple example pose graph with the nodes $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, and the three measurement constraints f_{12}, f_{13} and f_{23}.

In LLama-SLAM we use three main types of constraints we want to introduce briefly: The *Relative 3D Pose Constraint* represents a direct 3D pose measurement and the error function for this constraint is

$$\mathbf{e}_{R3,ij}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right) = \mathbf{T}_{ij}^{-1}\left(\widehat{\mathbf{T}}_{W,i}^{-1}\, \widehat{\mathbf{T}}_{W,j}\right), \tag{3}$$

where the $\mathbf{T}$s are homogeneous transformation matrices.

A GNSS measurement raises a *Global 2D Pose Constraint* with the 2D pose measurement $\boldsymbol{\tau}_{W,i} = [x\ y\ \theta]^{\mathrm{T}}$ containing the UTM coordinates x, y and the yaw θ. The error of a 3D pose which is constrained by such a measurement is computed as

$$\mathbf{e}_{G2,i}\left(\mathbf{x}_i = \widehat{\mathbf{T}}_{W,i},\ \mathbf{z}_i = \boldsymbol{\tau}_{W,i}\right) = \rho_{2D}\left(\widehat{\mathbf{T}}_{W,i}\right) - \boldsymbol{\tau}_{W,i}, \tag{4}$$

where ρ_{2D} is a function which extracts the 2D coordinates and the yaw from a 3D transformation matrix.

If a LLama j with position $\mathbf{t}_{W,j}$ is observed in the camera images, this introduces a *LLama Observation Constraint* with the error

$$\mathbf{e}_{L3,ij}\left(\mathbf{x}_i = \widehat{\mathbf{T}}_{W,i},\ \mathbf{x}_j = \mathbf{t}_{W,j},\ \mathbf{z}_{ij} = [\mathbf{p}_{ijL}, \mathbf{p}_{ijR}]^{\mathrm{T}}\right) = \left\|\pi_L\left(\widehat{\mathbf{T}}_{W,i}^{-1}\mathbf{t}_{W,j}\right) - \mathbf{p}_{ijL}\right\|_2 + \left\|\pi_R\left(\widehat{\mathbf{T}}_{W,i}^{-1}\mathbf{t}_{W,j}\right) - \mathbf{p}_{ijR}\right\|_2. \tag{5}$$

Therein $\mathbf{p}_L$ / $\mathbf{p}_R$ are the pixel coordinates of the LLama in the left/right image, and π_L / π_R are the projection functions for the left/right camera. We have chosen to minimize the re-projection error since this 2D-3D constraint should lead to a better accuracy than other constraint formulations [1].

B. LLama Quality Inference

The central part of LLama-SLAM is the assessment of the LLamas using probabilistic quality values. We define the random variable q_{lp} that states whether the LLama l is a high-quality LLama or not. The probability $\mathrm{P}\left(q_{lp} = 1\right)$ is thereby a useful quantity to rate and cull the LLamas. Since the visibility and appearance of a LLama will change depending on the viewpoint, $\mathrm{P}\left(q_{lp}\right)$ depends also on the camera position p. So for each LLama there is a local quality map, which holds the $\mathrm{P}\left(q_{lp}\right)$ values for different relevant viewpoints as shown in Fig. 3. Conveniently, the map thereby also implicitly stores possible viewing directions of a LLama. For simplification the viewpoint positions are assigned to $5\,\mathrm{m} \times 5\,\mathrm{m}$ cells on a rectangular grid.

The main property of high-quality LLamas is their long-term usability for localization, i. e. they can be easily detected and matched every time. Therefore the number of successful

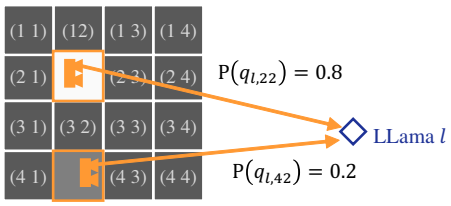

Fig. 3. For each LLama l we create a local LLama Quality Map that records the LLamas' quality from different viewpoint cells c_{lp} as probabilities. In this example the quality of the LLama from viewpoint cell (2 2) is high but low from cell (4 2).

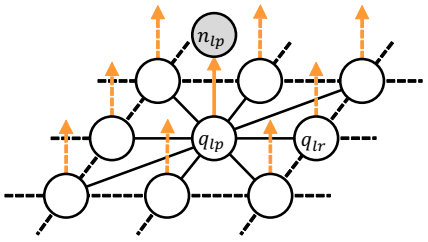

Fig. 4. Probabilistic graphical model that describes the relation between the quality values and the number of observations. The observation count n_{lp} is connected to the quality q_{lp} by the conditional $P(n_{lp}\,|\,q_{lp})$ (orange arrow). The quality values form a 2D Markov Random Field and are connected by similarity constraints $\psi(q_{lp}, q_{lr})$ (black edges). The figure shows only a cutout with one quality node and its neighboring nodes.

observations of a LLama is the definitive quantity to infer the quality of a LLama (c. f. Sec. I-B). Starting from this idea, we developed a probabilistic graphical model [18, p. 359ff] that connects the number of observations with the quality value. It also considers that the quality of a LLama from nearby viewpoint cells should be similar. These two assumptions lead to the model displayed in Fig. 4. Therein the number of successful observations n_{lp} is connected to the quality value by a conditional distribution (orange arrow). Observing or not observing the LLama is a repeated binary experiment and n_{lp} therefore follows the binomial distribution

$$P(n_{lp}\,|\,q_{lp}) = \binom{N_{lp}}{n_{lp}} \mu^{n_{lp}} \cdot (1-\mu)^{N_{lp}-n_{lp}}, \tag{6}$$

which is parameterized by $\mu = P(d_{lp}\,|\,q_{lp})$, i. e. the conditional probability of one successful observation, and the number N_{lp} of possible observations. To model the dependency between the viewpoint cells we consider the quality values q_{lp} as nodes in a Markov Random Field [18, p. 383ff]. Each node is therein connected to its eight neighbors via the potential function

$$\psi(q_{lp}, q_{lr}) = e^{-\alpha\,|q_{lp}-q_{lr}|}. \tag{7}$$

These undirected potential connections (black lines in Fig. 4) enforce similarity between the connected nodes.

To infer the quality values $P(q_{lp})$ from the observation counts n_{lp}, Loopy Belief Propagation [18, p. 417f] could be applied within this graphical model. However, to ease computation, we apply a modification, which Willert et al. already successfully applied to image denoising [19]: The 2D Markov Random Field is enhanced by another dimension with index k that represents the iterations. Each layer k within this model is a copy of the original 2D Markov Random Field and the similarity constraints are transformed to directed connections from the previous layer to the layer of the current iteration. This yields a 3D Bayesian Network without loops and the inference can therefore be carried out straightforward with standard message passing as specified below. For simplification we neglect the index l for the specific LLama in the following equations. The belief at node q_p^k in iteration k is given by

$$b\left(q_p^k\right) = \mathrm{m}_{n_p \to q_p^k} \cdot \prod_{q_r \in \mathrm{F}(q_p^{k-1})} \mathrm{m}_{q_r^{k-1} \to q_p^k}. \tag{8}$$

The first part is a message from the input node n_p with

$$\mathrm{m}_{n_p \to q_p^k} = P\left(n_p = \hat{n}_p\,|\,q_p\right). \tag{9}$$

Therein $\hat{n}_p$ is the actual number of successful LLama observations and $P(n_p\,|\,q_p)$ is given by (6). The second part is the product of the messages from each node q_r in the Markov blanket $\mathrm{F}\left(q_p^{k-1}\right)$ around the q_p^{k-1} node in the layer from the previous iteration. These messages are given by

$$\mathrm{m}_{q_r^{k-1} \to q_p^k} = \sum_{q_r^{k-1}=\{0,1\}} \psi\left(q_p, q_r^{k-1}\right) b\left(q_r^{k-1}\right) \tag{10}$$

and include the belief value from the previous iteration. In each iteration k of the inference the belief computation (8) is done for all q_p nodes in the map for the two possibilities $q_p = 0$ and $q_p = 1$. After normalizing these beliefs the inference can advance to the next iteration layer. The iteration is stopped if the beliefs do not change anymore. The final result of the iteration process is a LLama Quality Map given the current observation statistics. Three examples of inferred LLama Quality Maps are displayed in Fig. 10 and discussed in Sec. V-B. In the Sections IV-C and IV-D we will show how the LLama quality inference is embedded in our SLAM algorithm and how the results are used for LLama selection.

IV. LLama-SLAM Algorithm

In this section we explain how multiple sessions are processed to create the LLama Map. The input of the mapping process is the data from several recording sessions, where a session is a continuous recording without any interruptions. Before applying our algorithm, we feed the stereo image stream of each session to a visual odometry system (VO system) developed by Buczko et al. [20], [21]. Using the output of this VO system as additional input to our algorithm has two advantages: Firstly, we get reliable estimates for the camera movements between frames which are used to support the pose estimation. Secondly, we only use the inlier feature points from the VO movement estimation as LLama candidates. By utilizing this set of pre-filtered features, we can omit a feature detection stage and do not need to deal with problems caused by feature points on moving objects. Furthermore, we use features which already have proven to be useful for localization, which is favorable for LLamas as motivated in Sec. I-B.

Figure 5 provides an overview of the process to add a new session to the LLama Map. The sessions are processed one after another. First, the data from a session including

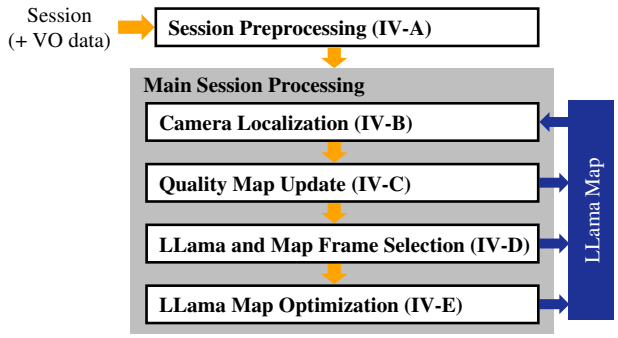

Fig. 5. Overview of the LLama-SLAM algorithm. For each step the subsection which describes the step in detail is given in parentheses.

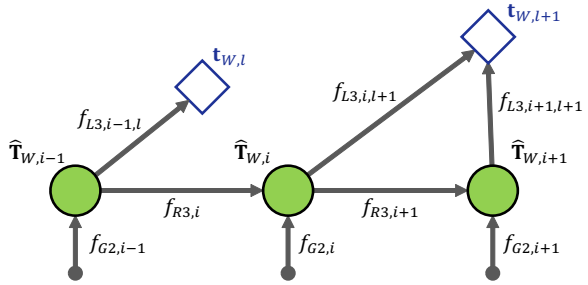

Fig. 6. In the camera localization step the camera poses $\widehat{\mathbf{T}}_{W,i}$ are optimized (green shaded nodes) as shown in this pose graph. They are connected to the nodes for the LLama positions $\mathbf{t}_{W,l}$ by observations constraints $f_{L3,il}$ which comprise the re-projection error. Furthermore, the constraints from the VO measurements $f_{R3,i}$ and the GNSS measurements $f_{G2,i}$ are incorporated.

the VO system output is preprocessed to refine the data. The main processing of a session starts with localizing the camera poses in the existing LLama Map. After the localization, the map is updated in three steps by first updating the LLama Quality Maps, then selecting LLamas and camera frames that will be added to the map and finally optimizing the updated LLama Map. All these steps of LLama-SLAM will be described in detail in the following sections.

A. Session Preprocessing

The first step in the algorithm is the preprocessing of the session data. Within the preprocessing first superfluous frames are removed, i. e. frames which have a similar position to other frames. After this selection process the frames are at least 2 m apart or differ by more than 20° in yaw rotation. Furthermore, individual feature point observations in the frames are combined to feature tracks, i. e. unique IDs are assigned to the VO feature points and kept over multiple consecutive frames. We also do a tentative camera pose optimization for the whole session using the GNSS measured world position $\boldsymbol{\tau}_{W,i}$ but also the relative VO pose estimations $\mathbf{T}_{i-1,i}$. By combining these constraints in the pre-optimization we can reduce the typical VO drift while also reducing errors of the GNSS measurements. The purpose of this tentative pose optimization is to start with good pose estimates in the localization step, which eases the LLama matching and serves as a convenient initialization for the pose graph optimization within the localization step.

B. Camera Localization

To update the LLama Map, we first need a precise localization of the camera poses of the current session in relation to the existing map. This is done by matching the VO-features with existing LLamas and optimizing the camera poses afterwards by minimizing the re-projection error. Since we know the complete feature track of a feature over several consecutive frames, the matching is not done frame by frame but rather by matching a whole feature track to one LLama. To achieve a successful match of a feature track with a LLama, the 3D position of the feature observations must be close to the LLama position $\mathbf{t}_{W,l}$ and the LLama's descriptor must be successfully matched in the left and right image of each frame belonging to the feature track. This matching criterion may seem quite restrictive but it ensures a low rate of false positive matches. Such false positive matches would corrupt the quality of our landmarks and must be avoided.

We use the ORB descriptor (Oriented FAST and Rotated BRIEF) developed by Rublee et al. [22] to describe the LLamas' visual appearance. Only the descriptor part of ORB, i. e. the Rotated BRIEF is used, since the feature detection is already done in the upstream VO system. We have chosen to use this descriptor since our long-term SLAM approach demands a descriptor with great robustness against appearance variations. With regard to this aspect the ORB descriptor has already proven its good performance in recent, successful methods like ORB-SLAM [23] or SOFT-SLAM [24]. To enhance the scale invariance of the descriptor we scale the patch, on which the descriptor is computed, according to the known distance to the camera.

After matching the landmarks with the VO feature tracks, we perform a pose graph optimization to get precise camera pose estimates. Therefore, we minimize the re-projection error of the matched LLamas while also trying to maintain consistency with the VO and GNSS measurements as shown in Fig. 6. The optimization problem in the localization step is therefore given by

$$\mathbf{X}^* = \underset{\mathbf{X}=\{\widehat{\mathbf{T}}_{W,i}\}}{\operatorname{argmin}} \sum_i \left(\sum_{l \in \mathrm{L}_i} f_{L3,il}\left(\widehat{\mathbf{T}}_{W,i},\ \mathbf{t}_{W,l},\ [\mathbf{p}_{ilL}, \mathbf{p}_{ilR}]^{\mathrm{T}}\right) + f_{R3,i}\left(\widehat{\mathbf{T}}_{W,i}, \widehat{\mathbf{T}}_{W,i-1}, \mathbf{T}_{i-1,i}\right) + f_{G2,i}\left(\widehat{\mathbf{T}}_{W,i}, \boldsymbol{\tau}_{W,i}\right) \right), \quad (11)$$

where we use the constraints defined in Sec. III-A. L_i is the set of observed LLamas from the camera pose i. If the LLama Map is used for online localization in a vehicle, only the process described in this section needs to be performed, possibly without the $f_{R3,i}$ constraints from the VO system.

C. Quality Map Update

With the estimated camera poses from the previous step, we are now able to assess the quality of the LLamas. As input for this inference we need to record for each LLama l the successful observations and the possible observations disjoint for the different viewpoint cells. To determine the number of possible observations N_{lp}, one has to count the visits of

each viewpoint cell c_{lp}, i. e. how many camera poses $\widehat{\mathbf{T}}_{W,i}$ lie inside this cell. The number of successful observations n_{lp} is determined by counting the number of instances a camera pose $\widehat{\mathbf{T}}_{W,i}$ lies inside cell c_{lp} and the LLama l is contained in the set of observed LLamas L_i. The resulting values are then added to the existing values of N_{lp} and n_{lp} and the likelihood (6) belonging to the n_{lp} node in the graphical model is updated. Using the inference method described in Sec. III-B, subsequently the new q_{lp} values inside the LLama Quality Maps are computed with the old map as initialization for the iteration process. These updated LLama Quality Maps are the key criterion for the following selection process.

D. LLama and Map Frame Selection

The selection process within LLama-SLAM is designed to assure a sparse map that also has a good spatial coverage. A sparse map is necessary for a good usability of the map in terms of computation, transmission and storage. Good spatial coverage ensures a high spatial availability of the localization and high accuracy. Furthermore it improves the localization performance, since a good coverage in 3D space should also lead to a good coverage in the image space. There is practical evidence [25] that a good distribution of landmarks in the image leads to more reliable pose estimates.

To ensure a sparse set of LLamas, we solely add LLamas with high quality to the LLama Map. Only LLamas for which $\left(\max_p q_{lp}\right) > Q_1$ is satisfied are added to the map, i. e. for at least one viewpoint the quality should be above the quality threshold Q_1. To check the spatial coverage we divide the map in the xy-plane in a grid of $20\,\mathrm{m} \times 20\,\mathrm{m}$ cells. After all new LLamas from a session are added, each cell is examined and if there are more than ten LLamas in one cell, only the ten LLamas with the highest quality are kept.

During the building process the LLama Map also contains a small number of image frames since they enable a more precise determination of the LLamas' spatial positions (see Sec. IV-E). For this purpose, we want to keep a spare set of high-quality map frames in the map, i. e. frames with many observations of high-quality LLamas. Consequently, the selection criterion for map frames is $(\sum_{l \in \mathrm{L}_i} q_{lp}) > Q_2$, i. e. the sum over the qualities of the observed LLamas in a frame i must be higher than the quality threshold Q_2.

This quality-based selection process to update the LLama Map is crucial to achieve updatability and dynamicity [3] within LLama-SLAM. It enables the algorithm to deal with lasting environmental changes. If new LLamas of high quality occur, they are added to the map, whereas permanently vanished landmarks are removed.

E. LLama Map Optimization

After a new session is added, the updated LLama Map is optimized. This is a crucial step within LLama-SLAM which provides precise estimations of the LLamas' positions by relying on the careful quality-based selection beforehand. Within this step, again a pose graph optimization is performed but now also the LLamas' positions $\hat{\mathbf{t}}_{W,l}$ are optimized and measurements from multiple sessions are combined. Figure 7 shows the corresponding pose graph and the optimization criterion is

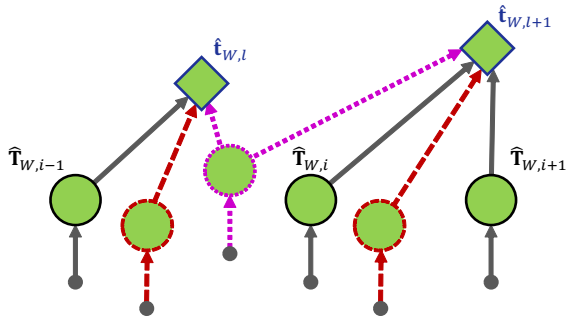

Fig. 7. In the pose graph for the map optimization, data from multiple sessions is combined. Here, nodes from three sessions are displayed (Session A —, Session B – –, Session C ····). The camera poses of the map frames as well as the LLama positions are optimized. This final optimization step is similar to a long-term Bundle Adjustment and allows to balance biases within the individual sessions.

$$\mathbf{X}^* = \underset{\mathbf{X}=\{\widehat{\mathbf{T}}_{W,i},\hat{\mathbf{t}}_{W,l}\}}{\operatorname{argmin}} \sum_i \left(\sum_{l \in \mathrm{L}_i} f_{L3,il}\left(\widehat{\mathbf{T}}_{W,i},\ \hat{\mathbf{t}}_{W,l},\ [\mathbf{p}_{ilL}, \mathbf{p}_{ilR}]^{\mathrm{T}}\right) + f_{G2,i}\left(\widehat{\mathbf{T}}_{W,i}, \boldsymbol{\tau}_{W,i}\right) \right). \quad (12)$$

Within the optimization each map frame is constrained by a set of LLama observations $f_{L3,il}$ and by a GNSS measurement constraint $f_{G2,i}$. The VO measurements are dropped, since the camera poses are already sufficiently constraint by the LLama observations. However, the $f_{G2,i}$ constraints are kept, since they keep the map frames and the LLamas in relation to a global reference. This allows to relate the resulting LLama Map to other available map data that is usually given in a global reference system.

The map optimization step can be seen as a long-term Bundle Adjustment where camera poses and LLama positions are optimized to fit selected measurements taken over a longer period of time. Because the measurements are taken at different days and months under different conditions, most of the contained biases are presumably uncorrelated and can be balanced by averaging them in the long run. In this way, LLama-SLAM kind of emulates a long-term GNSS measurement for each LLama and we assume that the LLama position estimates $\hat{\mathbf{t}}_{W,l}$ will converge towards their true positions if enough observations are evaluated.

V. Experiments

In this section, we present first results from a long-term mapping experiment to illustrate the capabilities of LLama-SLAM. Since our algorithm is focused on finding good persistent landmarks the following evaluation is also landmark-focused. The presented results are generated by mapping three sessions recorded on three different days in March and April 2017. Since this period covers spring season, there are some challenging changes in vegetation and weather conditions between the three sessions.

Fig. 8. LLama-SLAM can identify prominent persistent structures as long-term landmarks as illustrated by the two examples above. Each row shows the appearance of one LLama in three different mapping sessions. The green cross marks the LLama-position and the green square indicates the used descriptor patch size. A third example is displayed in Fig. 1.

Fig. 9. Points on varying structures are discarded as LLamas since they cannot be repeatedly observed. Three examples of discarded landmarks are displayed.

A. Identified LLamas

First, we present real-world structures that are identified by our algorithm as LLamas. In Fig. 1 and 8 three identified LLamas are displayed with their appearance over the three different sessions. We also present some counterexamples in Fig. 9. As intended, our algorithm uses prominent points on persistent structures like a curbstone, a sign or a street lamp as LLamas. Landmarks on varying structures like vegetation, parked cars or shadows are discarded.

B. Evolution of a LLama Quality Map

To illustrate how the quality update described in Sec. III-B and IV-C performs, the evolution of an example LLama Quality Map is visualized in Fig. 10. In this example the LLama is observable from the viewpoint cells (2 3), (3 3) and (4 3) in the first session (first column) with two successful observations per cell. As visible in the lower left heat map, this leads to quality values around 0.9 for the visited cells and approximately 0.75 for neighboring cells. More distant cells keep a value below 0.65. In the second session (heat maps in the second column) the LLama is observed from slightly different positions. The quality takes values close to 0.9 for all visited cells due to these further successful observations. Only the value of cell (2 4) is lowered to 0.25 since there was an unsuccessful observation. In the third and last session again a different route is taken. This time with even more unsuccessful observations. Therefore, the quality is drastically lowered in the visited cells. It is also lowered

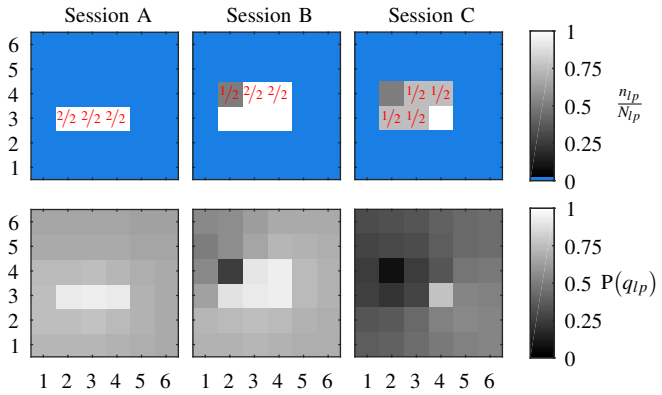

Fig. 10. Exemplary evolution of a LLama Quality Map over three mapping sessions. The first row shows the cumulative observations statistics as ratio of successful observations to possible observations. Not visited cells, i. e. cells with $N_{lp} = 0$, are colored blue. The red fractions show the observations statistics of the individual session. In the second row the LLama Quality Maps inferred from the cumulative observations statistics are displayed. While the quality rises in the first two sessions, it decreases in the third session, where the LLama could be observed in only one of two chances per cell. Please note how the probabilistic model enables reasonable estimations of the quality also for neighboring viewpoint cells which have not been visited.

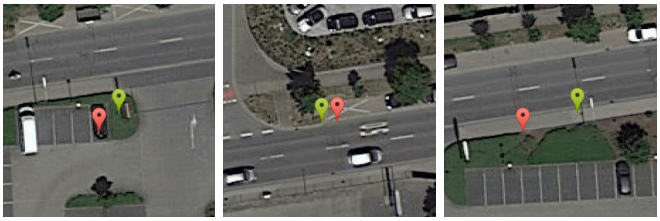

Fig. 11. LLama-SLAM utilizes GNSS measurements of the camera poses and is thereby able to determine the LLamas positions in a global referenced system. The aerial images show the estimated positions (red markers) of the LLamas introduced in Fig. 1 and 8. In comparison with the visible position in the images (green markers) an accuracy between 2 m and 7 m can be reached. (aerial images from Google Maps, © 2018, AeroWest)

in the neighboring cells due to the similarity constraints. Eventually this will lead to the deletion of this LLama since it has failed to provide long-term stability. This short example nicely illustrates how the LLama Quality Maps can represent possible viewpoints with associated quality and handle learning and oblivion of LLamas.

C. LLama Position Estimation

As stated in Sec. IV-E, LLama-SLAM is capable of estimating the LLamas' positions precisely in a global referenced coordinate system. Figure 11 shows the position of the LLamas introduced in Fig. 1 and 8 in relation to aerial imagery. The positions estimated by LLama-SLAM are quite close to the positions of the landmarks visible in the aerial images. This is a first indicator of the capabilities of the proposed method but we reckon greater accuracy can be achieved with further development of the algorithm.

VI. Conclusion

In this paper, we presented a new long-term mapping approach called LLama-SLAM, which focuses on identifying and mapping high-quality long-term visual landmarks. The core element of this algorithm is the inference of viewpoint

dependent quality probabilities for each landmark. These probabilities are inferred from observation statistics, whereupon the probabilities for different viewpoints are regarded as a Markov Random Field. The resulting LLama Quality Maps hold all necessary viewpoint information and eliminate the need to store camera pose information in the final map. Furthermore, they allow the selection of the LLamas with the greatest localization utility in a straight forward manner. The LLama Quality Maps are updated every time data from a new session is added. In combination with the quality-based selection, thereby LLama-SLAM achieves a long-term learning capability and can also deal with lasting environmental changes. It learns newly occurring LLamas and can forget permanently vanished LLamas. Since GNSS measurements of the camera poses are used in the map optimization steps, the precise positions of the LLamas can be determined in a global referenced system by averaging out the GNSS-errors from the individual sessions. This enables the vehicle to use or generate maps or share information with other vehicles since all information can be related to a common coordinate system. Such information could substantially aid advanced driver assistance systems or autonomous driving.

As next step in the development of LLama-SLAM, we are planning to integrate a more advanced LLama selection scheme to improve the spatial distribution of the landmarks, which increases their utility for localization tasks. Furthermore, we want to address the influence of changing lighting conditions on the LLama-matching. To a limited extend our procedure can handle this problem by favoring landmarks for which the ORB descriptor works in several lighting conditions. However, it would be more favorable to use a customized long-term descriptor to describe the visual appearance of the landmarks. The inclusion of other probabilistic quality measures is another enhancement we would like to investigate, e. g. incorporating stereo depth quality or the uncertainty of optical flow [26]. Furthermore, we plan to perform more extensive experiments by combining a greater number of sessions, trying different routes and mapping larger areas with overlapping routes. For a more comprehensive evaluation, we also intend to conduct high-precision reference measurements of exemplary vehicle routes and LLama positions and compare them to our mapping and localization results.

ACKNOWLEDGMENT

We kindly thank Continental for their great cooperation within Proreta 4, a joint research project of TU Darmstadt and Continental to investigate future concepts for intelligent and learning driver assistance systems.

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I: The first 30 years and fundamentals," *IEEE Robot. Automat. Mag.*, vol. 18, no. 4, pp. 80–92, 2011.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Transport. Syst.*, vol. 2, no. 3, pp. 194–220, 2017.

[4] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *Int. J. Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2014.

[5] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1156–1163.

[6] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4372–4378.

[7] T. Krajnik, J. P. Fentanes, J. M. Santos, and T. Duckett, "FreMEn: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 964–977, 2017.

[8] E. Johns and G.-Z. Yang, "Generative methods for long-term place recognition in dynamic scenes," *Int. J. Comput. Vision*, vol. 106, no. 3, pp. 297–314, 2014.

[9] M. Sons, M. Lauer, C. G. Keller, and C. Stiller, "Mapping and localization using surround view," in *IEEE Intelligent Vehicles Symp.*, 2017, pp. 1158–1163.

[10] F. Schuster, W. Zhang, C. G. Keller, M. Haueis, and C. Curio, "Joint graph optimization towards crowd based mapping," in *IEEE 20th Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–6.

[11] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, "Summary maps for lifelong visual localization," *J. Field Robotics*, vol. 33, no. 5, pp. 561–590, 2016.

[12] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale, "The gist of maps - summarizing experience for lifelong localization," in *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 2767–2773.

[13] F. Dayoub, G. Cielniak, and T. Duckett, "Long-term experiments with an adaptive spherical view representation for navigation in changing environments," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 285–295, 2011.

[14] L. Delobel, R. Aufrere, R. Chapuis, C. Debain, and T. Chateau, "Towards automated map updating for mobile robot localization," in *IEEE Intelligent Vehicles Symp.*, 2017, pp. 1342–1347.

[15] M. Stübler, S. Reuter, and K. Dietmayer, "A continuously learning feature-based map using a bernoulli filtering approach," in *Symp. on Sensor Data Fusion*, 2017, pp. 1–6.

[16] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.

[17] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3607–3613.

[18] C. M. Bishop, *Pattern recognition and machine learning*, ser. Information science and statistics. New York, NY: Springer, 2006.

[19] V. Willert and J. Eggert, "Belief propagation in spatiotemporal graph topologies for the analysis of image sequences," in *Int. Conf. on Computer Vision Theory and Applications*, 2010, pp. 117–124.

[20] M. Buczko and V. Willert, "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications," in *IEEE Intelligent Vehicles Symp.*, 2016, pp. 478–483.

[21] ——, "Monocular outlier detection for visual odometry," in *IEEE Intelligent Vehicles Symp.*, 2017, pp. 739–745.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE Int. Conf. on Computer Vision*, 2011, pp. 2564–2571.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[24] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *J. Field Robotics*, vol. 13, no. 2, p. 99, 2017.

[25] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," *IEEE Robot. Automat. Mag.*, vol. 19, no. 2, pp. 78–90, 2012.

[26] V. Willert and J. Eggert, "A stochastic dynamical system for optical flow estimation," in *IEEE 12th Int. Conf. on Computer Vision Workshops*, 2009, pp. 711–718.